

1. A method of debugging, comprising;
  - receiving a breakpoint selection for a program instruction associated with a first image file for a first processing engine;
  - identifying a source code file and a source code line in the source code file that
  - 5 generated the program instruction;
  - identifying further processing engines having an image file containing a program instruction generated by the source code line in the source code file; and
  - manipulating respective breakpoints for selected ones of the further processing engines based upon user selection, wherein the manipulated breakpoints correspond to
  - 10 program instructions generated by the source code line of the source code file.
2. The method according to claim 1, further including displaying a first screen that includes a mechanism to list the further processing engines.
- 15 3. The method according to claim 2, further including displaying a second screen to enable a user to select the further ones of the processing engines.
4. The method according to claim 3, further including displaying the second screen to enable the user to manipulate the breakpoints as one or more of inserting the breakpoint,
- 20 removing the breakpoint, enabling the breakpoint and disabling the breakpoint for the selected ones of the further processing engines.
5. The method according to claim 1, further including identifying the further processing engines by searching image files for the further processing engines to identify source code
- 25 files for the image files containing the source code line.
6. The method according to claim 1, further including receiving a selection of breakpoint type.

7. An article, comprising:

a storage medium having stored instructions thereon that when executed by a machine result in the following;

receiving a breakpoint selection for a program instruction associated with a first image file for a first processing engine;

identifying a source code file and a source code line in the source code file that generated the program instruction;

identifying further processing engines having an image file containing a program instruction generated by the source code line in the source code file; and

manipulating respective breakpoints for selected ones of the further processing engines based upon user selection, wherein the respective breakpoints correspond to program instructions generated by the source code line of the source code file.

8. The article according to claim 7, further including stored instructions to enable displaying a first screen that includes a mechanism to enable a user to see the further processing engines.

9. The article according to claim 8, further including stored instructions to enable displaying a second screen to enable a user to select the further ones of the processing engines.

10. The article according to claim 9, further including stored instructions to enable displaying the second screen to enable the user to select one or more of inserting the breakpoint, removing the breakpoint, enabling the breakpoint and disabling the breakpoint for the selected ones of the further processing engines.

11. A graphical user interface, comprising:

a first window to show microcode instructions associated with a first image file for a first processing engine in a processor simulator including a breakpoint for a first one of the microcode instructions generated by a source code line in a source code file;

5 a first menu to show user options including a first option to set breakpoints in multiple processing engines; and

a second window to display further processing engines having respective image files containing microcode instructions generated from the source code line and to enable the user to manipulate respective breakpoints for the further processing engines.

10

12. The graphical user interface according to claim 11, wherein the second window includes icons to manipulate the respective breakpoints by one or more of inserting breakpoints, removing breakpoint, enabling breakpoints and disabling breakpoints.

15 13. The graphical user interface according to claim 12, wherein the second window includes a display of the respective image files.

14. A debugger tool system, comprising

a processor; and

20 memory coupled to the processor, wherein the processor and memory combine to execute instructions that result in the following:

receiving a breakpoint selection for a program instruction associated with a first image file for a first processing engine;

25 identifying a source code file and a source code line in the source code file that generated the program instruction;

identifying further processing engines having an image file containing a program instruction generated by the source code line in the source code file; and

30 manipulating respective breakpoints for selected ones of the further processing engines based upon user selection, wherein the respective breakpoints correspond to program instructions generated by the source code line of the source code file.

15. The system according to claim 14, wherein manipulating the respective breakpoints includes one or more of inserting breakpoints, removing breakpoint, enabling breakpoints and disabling breakpoints.

5 16. The system according to claim 14, further including displaying a list of the identified further processing engines.

17. A network forwarding device, comprising:

a switch fabric; and

10 multiple line cards interconnected by the switch fabric, individual ones of the line cards comprising a network processor including multi-threaded microengines configured to execute microcode,

wherein the microcode includes instructions developed using a debugger tool that identified and enabled breakpoints to be set on multiple ones of the multi-threaded

15 microengines, wherein the breakpoints correspond to a common line of source code in a common source code file.

18. The device according to claim 17, wherein the debugger tool includes a graphical user interface comprising:

20 a first window to show microcode instructions associated with a first image file for a first processing engine in a processor simulator including a breakpoint indicator for a first one of the microcode instructions generated by a source code line in a source code file;

a first menu to show user options including a first option to set breakpoints in multiple processing engines; and

25 a second window to display further processing engines having respective image files containing microcode instructions generated from the source code line and to enable the user to manipulate respective breakpoints for the further processing engines.

19. The device according to claim 17, wherein the breakpoints were manipulated by one or more of inserting the breakpoints, removing the breakpoints, enabling the breakpoints and disabling the breakpoints for the selected ones of the further processing engines.